Linear Regression Models and Neural Networks for the Fast Emulation of a Molecular Absorption Code

Guillaume Euvrard, Isabelle Rivals, Thierry Huet, Sidonie Lefebvre, Pierre Simoneau

Linear regression models and neural networks for the fast emulation of a molecular absorption code

Guillaume Euvrard,[1,*] Isabelle Rivals,[1] Thierry Huet,[2]
Sidonie Lefebvre,[2] and Pierre Simoneau[2]

[1]Equipe de Statistique Appliquée, École Supérieure de Physique et de Chimie Industrielles,
10 rue Vauquelin, 75005 Paris, France

[2]Département d'Optique Théorique et Appliquée, Office National d'Etudes et de Recherches Aérospatiales,
Chemin de la Hunière, 91761 Palaiseau Cedex, France

*Corresponding author: guillaume.euvrard@espci.fr

Received 22 July 2009; revised 16 October 2009; accepted 23 October 2009;
posted 29 October 2009 (Doc. ID 114604); published 3 December 2009

The background scene generator MATISSE, whose main functionality is to generate natural background radiance images, makes use of the so-called Correlated K (CK) model. It necessitates either loading or computing thousands of CK coefficients for each atmospheric profile. When the CK coefficients cannot be loaded, the computation time becomes prohibitive. The idea developed in this paper is to substitute fast approximate models for the exact CK generator; using the latter, a representative set of numerical examples is built and used to train linear or nonlinear regression models. The resulting models enable an accurate CK coefficient computation for all the profiles of an image in a reasonable time. © 2009 Optical Society of America

OCIS codes: 000.4430, 010.1030, 010.5620, 200.4260, 010.1300, 110.2960.

1. Introduction

Optronic sensor designers require the calculation of the radiation contrast between targets and background in order to assess the detection performance of their surveillance systems. In addition, since the sensor may be used in many different meteorological conditions, the background radiance and the transmitted radiation of the target must be evaluated for a large set of atmospheric conditions. Advanced Modeling of the Earth for the Imaging and the Simulation of the Scenes and their Environment (MATISSE-v1.5) [1], whose main functionality is to generate natural background radiance images and useful atmospheric radiative quantities (radiance and transmission along a line of sight, local illumination, or solar irradiance, for example), is hence a perfectly suited tool for performance estimation of optronic sensors. Unlike most other image simulators, which favor computation speed to the detriment of physical realism, MATISSE is developed to generate reference images using efficient methods in terms of accuracy and computation time.

Therefore, molecular absorption calculation, required to solve the radiative transfer equation, is made in MATISSE using a Correlated K (CK) model [2,3]. This model has two benefits: its numerical efficiency and computational speed in modeling nongray absorption by gases in inhomogeneous atmospheres, and its ability to include atmospheric multiple scattering computation at a lower cost than using line-by-line calculations. Prior to running MATISSE, each atmospheric profile describing the evolution of the pressure, the temperature, and the molecular mixing ratios on an altitude grid must be converted to a CK profile by the use of a CK generator. CK parameters are computed for each altitude within the user-required spectral band (from 700 up to $25,000\,\mathrm{cm}^{-1}$; a resolution of $1\,\mathrm{cm}^{-1}$ allows the continuous reconstitution of the band) and stored in files. These files are then read by MATISSE and

0003-6935/09/356770-11$15.00/0

introduced into the radiative transfer equation. Once the CK coefficients have been set, running MATISSE is very efficient since the same coefficients can be associated with various sight conditions.

The main drawback of this approach lies in the computation time of the CK coefficients whenever a new profile has to be introduced in MATISSE (typically a radiosounding). The CK calculation is indeed based on a very time-consuming line-by-line model [4] in order to take into account the variability of the absorption coefficient versus the wavenumber. In practice, it necessitates approximately 10 min for one single profile, and in order to account for the large number of thermodynamic profiles that is required for atmospheric variability modeling [for example, to build a three-dimensional (3D) scene whose profiles come from weather forecast output], the overall CK computation would take at least two months.

As accelerating simplification, current radiative transfer models [5] make use of one-dimensional (1D) stratified atmospheres, but they are, therefore, not able to take into account the horizontal structure of the atmosphere. Attempts have recently been made [6] to introduce this dependency by means of interpolations and scaling factors. In MATISSE, the 3D variability of the aerosols is already handled by the Global Aerosol Dataset (GADS) [7]. If an alternative, faster, and, nevertheless, accurate CK computation can be found, we will be able to both deal with the 3D variability of the atmosphere and preserve the reference side of radiative transfer calculus made by MATISSE.

In this paper, we suggest substitution with approximate models, expected to be much faster to run, for the exact CK generator. The idea is to build, for each wavenumber, a local model describing the dependency of each particular CK coefficient on the thermochemical conditions from a large number of atmospheric profiles. First, the CK generator would be used to compute a representative set of numerical examples of the atmospheric conditions and of the corresponding CK values; these examples would then be used to fit linear or nonlinear regression models. Provided that their domain of validity encompasses the variability of the profiles, the models will enable us to deal with 3D scenes made up of about 10,000 profiles describing the state of the atmosphere above a particular geographic region. The objective of this paper is, hence, to describe how statistical modeling can replace the CK generator, with significant reduction of computation time and small loss of accuracy.

The paper is organized as follows. Section 2 describes the CK coefficients, while Section 3 is dedicated to the data set. Section 4 details the algorithm that builds the models. Finally, the resulting models and their performance are presented, in terms of accuracy as well as of computation time.

2. Correlated K Description

The integration of any radiative function $V(k)$ on the wavenumber interval $\Delta\sigma$ using a line-by-line method is inconceivable, due to the variability of the absorption coefficient $k(\sigma)$ on the interval $\Delta\sigma$; a reliable result would imply extremely small integration steps, and would be prohibitive to compute. However, by sorting the absorption spectrum, it is possible to substitute integration on the absorption coefficient to the wavenumber integration. Thus, the estimation of the mean value of V requires only the values taken by the absorption coefficient k on the interval $\Delta\sigma$, together with their occurrence number; the exact correspondence between the wavenumber σ and the absorption coefficient k is not required.

The mean value on the interval $\Delta\sigma$ of the radiative function $V(k(\sigma))$ can then be written as

$$\bar{V}_{\Delta\sigma} = \frac{1}{\Delta\sigma}\int_{\Delta\sigma} V(k(\sigma))\mathrm{d}\sigma = \int_0^1 V(k(g))\mathrm{d}g,$$

where the function $k(g)$ describes the cumulative distribution of the k values; for each ratio g, the value $k(g)$ is such that the ratio occurrence of the event "$k \le k(g)$" equals g (see Fig. 1).

This integral can be approximated by a Gauss quadrature: by choosing N quadrature points

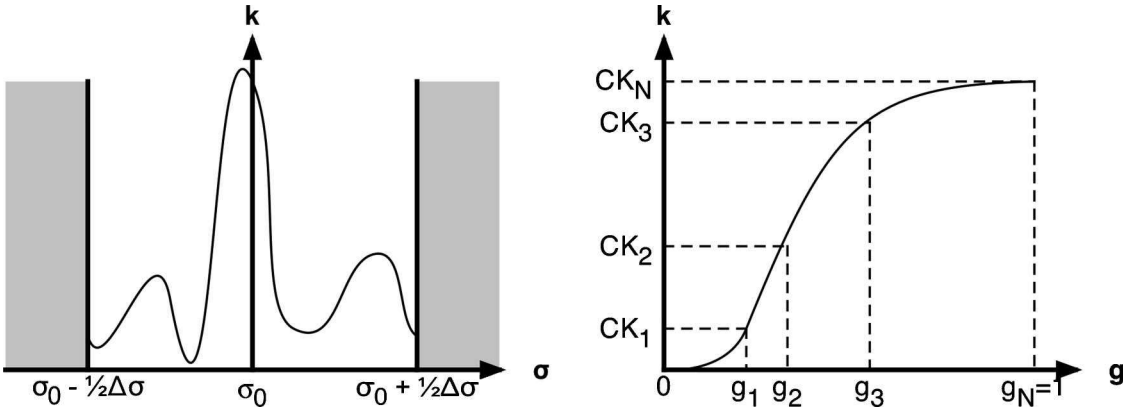

Fig. 1. The dependency of the absorption coefficient k on the wavenumber σ (left) can be replaced by the cumulated frequencies of the k values (right). The CK_i coefficients are the values of the absorption coefficient for specified values g_i of the occurrence ratio.

$\{g_i, i = 1, \ldots, N\}$ and defining CK_i as the $k(g)$ values to the quadrature points

$$CK_i = k(g_i),$$

it can take the form

$$\bar{V}_{\Delta\sigma} = \int_0^1 V(k(g))\mathrm{d}g \approx \sum_{i=1}^{N} w_i V(\mathrm{CK}_i). \qquad (1)$$

In MATISSE, the number of quadrature points has been set to 17, the 17 occurrence ratios $\{g_i, i = 1, \ldots, 17\}$ being fixed, as well as the 17 weights w_i. This method enables us to suppress the explicit dependency on the wavenumber on the spectral interval $\Delta\sigma$.

In fact, the absorption coefficient k depends not only on the wavenumber σ, but also on the thermophysical conditions. More precisely, the absorption coefficient k is a function of the wavenumber σ, the temperature T, the pressure P, and the mixing ratios X_m of each molecule $m (m = 1, \ldots, M)$. Consequently, the k distribution and the CK_i values also depend on these variables and vary with the position considered in the atmosphere:

$$\mathrm{CK}_i = \mathrm{CK}_i(\sigma_0, P, T, X_1, \ldots, X_M).$$

A large part of the computations of MATISSE is dedicated to the CK_i estimations in the area of interest. It consists in replacing the thermophysical data, represented by the $(P, T, X_1, \ldots, X_M)$ values at different points in the atmosphere, by optical parameters, represented by the 17 CK_i values at these points and at each spectral resolution element σ_0 (we consider the band from 700 up to $25,000\,\mathrm{cm}^{-1}$ with a $1\,\mathrm{cm}^{-1}$ resolution).

When this computation has been done, the integration of any radiative parameter (atmospheric source functions, extinction coefficients, background radiance) is performed rapidly; the parameter values are computed in the whole area for each of the 17 CK values. Image generation then consists of loading or computing the CK coefficients and, for each pixel, propagating the source terms corrected by molecular absorption by using Eq. (1). A typical illustration is given by Eq. (2). The left term is the average radiance seen by the observer in the spectral interval $\Delta\sigma$ and the θ, φ direction. The first term on the right represents the surface background radiance (land, sea, or cloud) propagated through the atmosphere. The second term on the right represents the contribution of atmospheric source functions. Once computed, the $L^i{}_S(\theta, \varphi)$ and $J^i(s, \theta, \varphi)$ terms are stored in files. Radiance computation then consists of reading these radiative quantities and propagating them with the CK model along the path from the background to the observer:

$$\bar{L}^{\Delta\sigma}_{obs}(\theta, \varphi) = \sum_{i=1}^{N} w_i L^i_S(\theta, \varphi) \exp\left(-\int_{s_0}^{\mathrm{obs}} k_i^{\mathrm{ext}}(s')\mathrm{d}s'\right) + \sum_{i=1}^{N} w_i \int_{s_0}^{\mathrm{obs}} w_i \exp\left(-\int_{s}^{\mathrm{obs}} k_i^{\mathrm{ext}}(s')\mathrm{d}s'\right) \times J^i(s, \theta, \varphi) k_i^{\mathrm{ext}}(s)\mathrm{d}s, \qquad (2)$$

where

- N is the number of quadrature points, $N = 17$ in MATISSE;
- w_i is the ith quadrature weight;
- $k^{\mathrm{ext}}i(s)$ is the ith value of the CK parameters at location s: $k_i^{\mathrm{ext}}(s) = k_i(s) + k^{\mathrm{ext}}(s)$ with $k_i(s)$ one of the N CK values calculated at point s and $k^{\mathrm{ext}}(s)$ the absorption coefficient of the aerosols at location s;
- $L^i{}_s(\theta, \varphi)$ is the background surface radiance in the (θ, φ) direction when the radiative transfer equation is solved with the ith CK parameter; and
- $J^i(s, \theta, \varphi)$ is the atmospheric source function, at location s and in the (θ, φ) direction, when the radiative transfer equation is solved with the ith CK parameter.

Calculation of the brightness mean value amounts to N independent calculations, and a weighted sum of their results.

When the average radiance is calculated with the CK method, the atmospheric source functions have to be determined, as well as the ground and cloud radiances for each of the N values $k^{\mathrm{ext}}{}_i(s)$. This method makes it possible to couple the molecular absorption with the scattering by way of the $L^i{}_S(\theta, \varphi)$ and $J^i(s, \theta, \varphi)$ parameters, and also to use Beer's law, as can be seen in Eq. (2).

This mode of computation is based on the conversion, for each profile, of its thermophysical representation (pressure, temperature, and molecular mixing ratios) into its optical characterization—a set of CK values defined for each spectral resolution element σ_0. It allows fast computations for all the radiative parameter integrations, so that the computation time is essentially devoted to this task. But the latter is highly time consuming: for a profile, that is, for an atmospheric column above a point on Earth discretized in 49 altitude points, the 24,300 wavenumbers × 49 altitudes × 17 quadrature points = 20,241,900 CK computations require 10 min. For a whole 3D scene including 10,000 profiles, the computation time would be about two months. The idea studied in this paper is to accelerate the CK computations by using approximate but faster models.

An estimation of the computation sensitivity to the CK accuracy has suggested that, for each profile, each wavenumber and each altitude, only the first eight nonzero CK coefficients need to be approximated with a relative error of less than 5% or an absolute error of less than $10^{-12}\,\mathrm{cm}^{-1}$.

3. Dataset

The dataset *climato* [8] consists of 4672 profiles. A profile represents a column above a given point on Earth, on a given date. For each profile, the record includes the thermodynamic and optical properties at 49 altitudes defined by 49 discrete values of the pressure (see Fig. 2). These levels of pressure run from 10^{-2} up to 1.03×10^5 Pa.

The thermodynamic representation consists of the pressure, the temperature, and the molecular mixing ratios of 31 of the main molecules represented in the HITRAN database [9], later denoted by the vector $\mathbf{X}$. The optical representation consists of the CK_i values $(i = 1, \ldots, 17)$ for each wavenumber. In the future, a spectral bandwidth from 700 to $25,000\,\mathrm{cm}^{-1}$ with a resolution of $1\,\mathrm{cm}^{-1}$ will be needed, leading to $24,300 \times 49$ pressures $\times$ 17 quadrature points $= 20,241,900$ CK values. But for this study, which is meant as a preliminary validation of the proposed approach, the coefficients were computed for a subset of 198 spectral values in the bandwidth from 750 to $3,015\,\mathrm{cm}^{-1}$, leading to $198 \times 49 \times 17 = 164,934$ CK coefficients, and only those are to be modeled.

The profiles have been chosen to cover a wide set of thermodynamic conditions. The geographical location runs from latitude −90° up to +90° with a 2.8125° sampling interval, each of them during a whole year, with a five-day sampling interval. This results in a dataset of 64 latitudes × 73 dates $= 4,672$ profiles.

In addition to the profiles, the dataset also contains, for each wavenumber value, the list of the molecules that absorb at this frequency; their corresponding components in the vector $\mathbf{X}$ are likely to be pertinent variables of the models.

Finally, we would like to stress that these data have been obtained from computations and have no measurement error. Thus, the output variability is due only to the input variability and the objective for the model accuracy is not limited by any noise standard deviation.

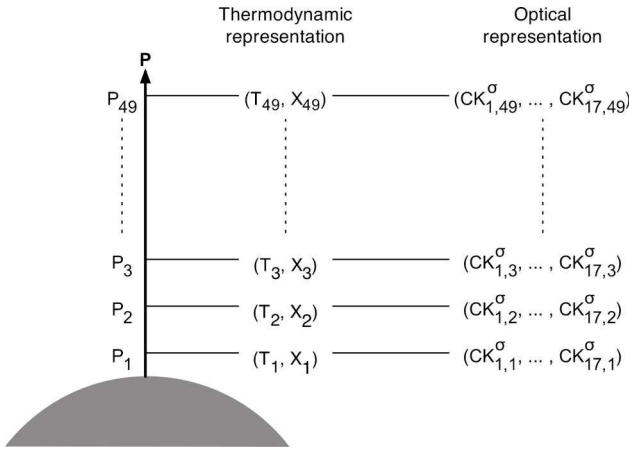

Fig. 2. A profile is a column above a given point on Earth at a given time. In the database, it is represented at discrete levels of the pressure, which determine a set of altitudes. The thermodynamic representation includes, at each altitude, the value of the temperature T and of the molecular mixing ratios vector $\mathbf{X}$. The optical representation includes, at the same altitudes, the CK_i values $(i = 1, \ldots, 17)$ for each wavenumber σ. The database *climato* contains 4672 profiles.

4. Construction of the Models

The CK coefficient values depend on the wavenumber, the pressure, and the other thermodynamic conditions. However, with the wavenumber and the pressure being discretized with small steps, we felt that simple local models for each wavenumber and each pressure value would be more appropriate than a large and complex model having to be valid in the whole spectral and pressure domain. Thus, for each quadrature point i, each wavenumber σ, and each pressure value P_h, the relation

$$(T_h, X_h) \rightarrow \mathrm{CK}_{i,h}^{\sigma}$$

has to be fitted by a local model. The number of models is, hence, the number of CK coefficients; that is 164,934 today with the considered wavenumber subset, and will be 20,241,900 in the future. The procedure has to be completely automated; it is not possible to prompt for a user choice at each model construction.

A. General Framework

The construction of a model amounts to selecting an accurate function among a given set of candidates. The function sets considered here are polynomials and neural networks. As a matter of fact, both are universal approximators; a polynomial or a neural network can approximate any function to a specified accuracy, provided that the degree and the number of monomials, or the number of neurons, is large enough. However, a crucial issue that the algorithm has to address is the choice of the *whole* structure; for a polynomial, what should be its degree and which monomials should it include? And for a neural network, how many neurons should it contain?

When the whole structure is given, the model is merely a function $f(\mathbf{x}, \boldsymbol{\theta})$ with an input vector $\mathbf{x} =$ (temperature T, molecular mixing ratio vector $\mathbf{X}$) and a parameter vector $\boldsymbol{\theta}$. The remaining question is then, what value should the parameter vector $\boldsymbol{\theta}$ take? A part of the dataset, called the *training* set, is used to compute a mean square error of the model, and the parameter vector is estimated as the least squares solution:

$$\boldsymbol{\theta}_{\mathbf{ls}} = \mathrm{argmin}_{\theta} \sum_{\mathrm{ex}} [y(\mathrm{ex}) - f(x(\mathrm{ex}), \theta))]^2. \qquad (3)$$

Once different structures have been trained, they can be compared. But one cannot simply choose the model with the smallest mean squared error on the training examples. The latter best fits the examples used to estimate its parameters, but it is not necessarily the model that best fits the function, due to overfitting. To detect overfitting, it is necessary to measure the errors on examples that have not been

Table 1. Performance Measures

Error Type	Training Set	Validation Set	Test Set
Root mean square error: rMSE	rMSE-Tr	rMSE-V	rMSE-Ts
Max error: MXE	MXE-Tr	MXE-V	MXE-Ts
Root mean square relative error: rMSRE	rMSRE-Tr	rMSRE-V	rMSRE-Ts
Max relative error: MXRE	MXRE-Tr	MXRE-V	MXRE-Ts

used to estimate the parameters. Thus, another set of examples has to be considered, the *validation* set; it is used to compare two models, the parameters of which have been previously estimated with the training set.

Provided that a training set and a validation set are available, an algorithm can run through different model structures until an accurate one is encountered; for each structure, the parameters are taught to fit the training examples and the resulting model accuracy is then evaluated on the validation set. The final model is the first accurate structure encountered (evaluated on the validation set) with the optimal parameters (evaluated on the training set). Its global performance can be estimated, but on neither the training set, since it has been used to evaluate the parameters, nor on the validation set, since it has been used to select the structure. This is the reason why a third set of examples is considered: the *test* set. This set is not at all used in the algorithm that builds the models. The test errors are only computed at the end of the algorithm, in order to evaluate how the final model fits the function. The initial dataset is hence divided into three subsets; here they are chosen of equal size.

The next paragraph introduces the main measures of accuracy and the second paragraph describes the algorithm that explores the model structures and selects the final model.

B. Error Functions

For each CK coefficient and for each example ex, both an absolute and a relative error can be computed:

$$\begin{cases} E(ex) &= \mathrm{CK}_{\mathrm{model}}(ex) - \mathrm{CK}_{\mathrm{real}}(ex) \\ RE(ex) &= \frac{\mathrm{CK}_{\mathrm{model}}(ex) - \mathrm{CK}_{\mathrm{real}}(ex)}{\mathrm{CK}_{\mathrm{real}}(ex)} \end{cases}.$$

For a whole example set, these two errors can be described both by their root mean squares (rMSE and rMSRE) and by their maximum absolute values (MXE and MXRE). Finally. the three example sets lead to 12 error functions for a single model (Table 1).

The accuracy specified for the CK estimations depends only on the first eight nonzero coefficients. Therefore, the relative errors are computed with the corresponding examples only. For each relation, the mean relative errors are computed with the profiles for which the CK coefficient is one of the first eight nonzero CK (i.e., larger than $10^{-12}\,\mathrm{cm}^{-1}$). The data being noiseless, these errors are representative of the approximation errors and are reliable to decide whether a model satisfies the specification or not; in the procedure, the accuracy specification is fulfilled if the model has either a rMSE in validation (rMSE-V) of less than $10^{-12}\,\mathrm{cm}^{-1}$ or a rMSRE in validation (rMSRE-V) of less than 5%. Such a model is considered as accurate enough and, when encountered, stops the procedure. The model comparison is done via their mean square error in validation (MSE-V).

C. Modeling Procedure

1. *Overview*

The procedure follows the general scheme described in [10] and shown in Fig. 3. After setting the candidate inputs—the temperature and the mixing ratios of the molecules that are optically active at the considered wavenumber—it builds a polynomial model with a small degree. This step is fast and results, most of the time, in a sufficient model. If not, the procedure builds a neural network whose inputs are those of the polynomial. This step is time consuming, but it is performed only for the few most complex relations, for which it is worth it. If the neural network is accurate enough, it is chosen as the final model. If not, that is, if neither the polynomial nor the neural network fulfills the accuracy specification, both are compared and the model with the lowest MSE-V is selected as the final one.

The two next subsections detail the polynomial and neural network construction.

2. *Polynomial Construction*

A polynomial structure is a set of monomials and its parameters are their multiplicative coefficients. The

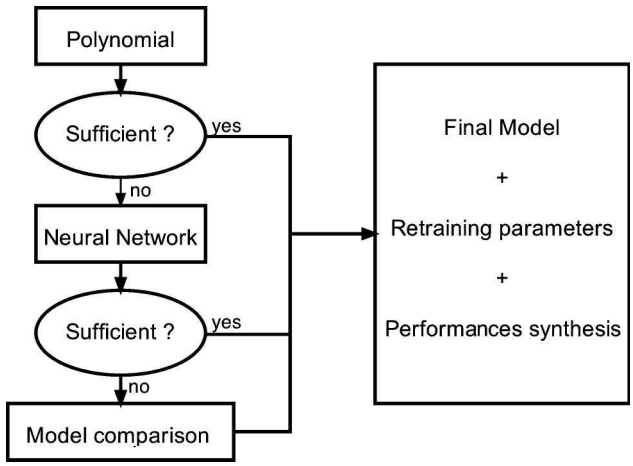

Fig. 3. General scheme of the modeling procedure.

model depends linearly on these parameters and the solution $\boldsymbol{\theta}_{ls}$ of Eq. (3) is given by the ordinary least squares formula.

The remaining issue of the polynomial construction is to decide which monomials it should contain. In the procedure, the maximum degree is set to 3, so that the possible monomials consist of those depending on the candidate inputs and having an adequate degree. Among these possible monomials, a selection has to be done. To start with, the procedure reads the desired output vector $\mathbf{Y}_d$, made of the training example outputs, and computes the experience matrix $\mathbf{X}$, whose columns correspond to the possible monomials and whose rows are the monomial values at the training examples.

Then the procedure uses the Gram–Schmidt orthogonalization [11] to rank the monomials in order of their decreasing contribution to the output's explanation. First, the correlations of all the columns of the matrix $\mathbf{X}$ with the output $\mathbf{Y}_d$ are computed; the most correlated column is placed in first position, as well as the monomial it corresponds to. The remaining columns of the matrix $\mathbf{X}$ and the vector $\mathbf{Y}_d$ are then projected orthogonally to the first column, and the correlations of the resulting columns to the resulting vector are computed. The most correlated column, as well as the corresponding monomial, are placed in the second position. The same computation is iterated until all the columns of the matrix $\mathbf{X}$ and all the monomials have been sorted. In the end, the matrix $\mathbf{X}$ is orthogonal; its columns and the monomials are sorted according to their contribution to the model performance. Thanks to the orthogonal projections, this sorting eliminates the redundancies; at each iteration, a column is ranked and all the remaining ones are projected orthogonally, so that the information they have in common with the selected column is deleted before further ranking. Thus, if two monomials contain redundant information on the output, only one of them is ranked at a good position.

The resulting rank defines a sequence $f_k(\mathbf{x}, \boldsymbol{\theta})$ of nested model structures, the first one being the constant model, the second one including the constant and the first monomial, and so on: let $\varphi_k(\mathbf{x})$ denote the monomial sorted at the position k; then the model $f_k(\mathbf{x}, \boldsymbol{\theta})$ is

$$f_k(\mathbf{x}, \boldsymbol{\theta}) = \theta_0 + \sum_{i \le k} \theta_i \varphi_i(\mathbf{x}). \tag{4}$$

The final polynomial structure is an element of this sequence and the remaining question is the choice of the size k. The larger is k, the richer the model, but also the higher the risk of overfitting.

To select it, the procedure increases it from 0 (the model is a constant) to its maximum (the total number of possible monomials). For each value of k, the parameters are estimated on the training set and the accuracy on the validation set. The procedure stops when the model $f_k(\mathbf{x}, \boldsymbol{\theta}_{ls})$ is accurate enough, or when the mean square error in MSE-V is not lower than with $k-1$ monomials, or when k reaches its maximum. If the final polynomial model fulfills the accuracy specification, it is set as final model. If not, a more complex one has to be considered and the procedure builds a neural network.

Polynomials are good approximators and their training is fast, but when the number of inputs and the maximum degree increase, the number of possible monomials grows drastically, resulting in very high computation time and memory request. For instance, with 15 inputs and a maximum degree of 3, the number of monomials is 816, but when the maximum degree is set to 4, 5, or 6, the number of monomials goes up to 3876, 15,504 or 54,264, respectively. Thus, when small degree polynomials are not accurate enough, they need to be replaced by models whose complexity can be increased more gradually. In that case, neural networks are introduced.

3. *Neural Network Construction*

The neural networks considered here are networks with one hidden layer (Fig. 4). Each neuron i of the hidden layer computes

$$\begin{cases} \text{Potential}: & V_i = w_{i0} + \sum_j w_{ij} x_j \\ \text{Output}: & O_i = \tanh(V_1) \end{cases},$$

where w_{ij} is the weight of the connection from the input j to the neuron i. The output of the network is given by

$$O = w_0 + \sum_i w_i O_i,$$

where the w_i are the weights of the connections from the hidden neurons to the output.

Only the inputs that were included in the final polynomial model are fed to the neural network. The neural network construction consists in setting the number n_{hid} of hidden neurons and the parameter vector $\boldsymbol{\theta}$ whose components are the weights w_{ij} and w_i.

When the number n_{hid} is defined, the model is a function $f(\mathbf{x}, \boldsymbol{\theta})$ of the input vector $\mathbf{x}$ and of the parameter vector $\boldsymbol{\theta}$. As for the polynomial models, the

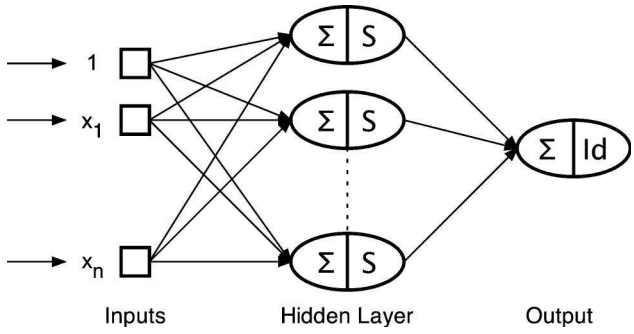

Fig. 4. Neural network with one hidden layer.

training phase consists in finding the $\boldsymbol{\theta}$ value that minimizes the square error on the training set, that is, in solving Eq. (3). But, in this case, the network output $f(\mathbf{x}, \boldsymbol{\theta})$ is not linear with respect to the parameters and there is no direct formula for the optimal $\boldsymbol{\theta}_{ls}$. The latter has to be estimated with an iterative procedure that is expected to converge to the optimum. In this work, we use the Levenberg–Marquardt algorithm, which, at each iteration, makes a variation $\Delta\boldsymbol{\theta}$ of the parameter that depends on the gradient and on an approximation of the Hessian of the square training error [12].

To set the hidden layer size n_{hid}, the procedure operates in a manner similar to that for the polynomial size: it gradually increases n_{hid} and, for each value, estimates the least squares parameter $\boldsymbol{\theta}_{ls}$ on the training set and the performance on the validation set. The procedure stops when the model fulfills the accuracy specification, or when the MSE-V is not lower with n_{hid} neurons than with $n_{\text{hid}} - 1$, or when a maximum value has been reached.

If the neural network is accurate enough, it is selected as the final model. If not, its MSE-V is compared to that of the polynomial, and the better of the two is the final model.

As compared to polynomials, neural networks are parsimonious; they can approximate complex functions with fewer parameters. In brief, this property is due to the fact that the basis elements (the neurons) can be chosen in a nonlinear fashion [13,14]. Furthermore, the model complexity can be increased more gradually: when the numbers of inputs and of hidden neurons grow, the number of parameters grows like their product and there is no gap as there is when increasing the polynomial degree. But the drawback is the large computation time needed for the neural network training.

5. Results

The procedure described above has been applied to the 198 wavenumbers, resulting in 198×49 pressure values $\times$ 17 CK coefficients $= 164,934$ models. This section describes them, in terms of accuracy, size, and computation time.

A. Model Structure and Accuracy

For nearly two-thirds of the relations (102,185 among the 164,934), a constant model proves to be sufficient. This is the case for the CK coefficients whose values are always less than $10^{-12}\,\text{cm}^{-1}$ (as it happens to be for the first CK coefficients at high altitudes), and for the CK coefficients which, for every example, have eight nonzero CK coefficient before them.

Hence, there are 62,749 nontrivial relations left. For only 33 of them, the procedure could not build a sufficient model. For all the 62,716 other relations, a model has been found with a rMSE-V set of less than $10^{-12}\,\text{cm}^{-1}$ or with a rMSRE-V set of less than 5%. To be more precise, for 62,598 relations, the procedure has built a polynomial model fulfilling the accuracy specification. Then, for the 151 other relations, the polynomial is not sufficient and the procedure has built a neural network. In 118 cases, it indeed succeeded in building a sufficient model while, in 29 other cases, it built a model that is not sufficient but still has better performance than the polynomial. The remaining four models could not be improved by a neural network (see Table 2). All the polynomial models have been built in 24 h on a single computer (Dual Core AMD Opteron 2212 processor, 2,000 MHz). For the neural models, one week on four similar computers was needed.

For the 62,749 nontrivial models, a rMSRE-V and a MXRE-V can be computed on the validation set. Their means are 3.3% for the rMSRE-V and 18.1% for the MXRE-V, and their distributions are plotted in Fig. 5. The same indicators can be computed on the test set. Their means equal 3.3% and 12.3%, respectively, and their distributions are plotted in Fig. 6.

Computed on the validation or on the test set, the distributions of the rMSRE are very similar; their means are identical (3.3%) and their histograms have the same shape (left part of Figs. 5 and 6). In particular, a "threshold" value at 5% can be noted: the root mean square error values under 5% are much more numerous than those above 5%. For the errors on the validation set, this is a direct consequence of the building procedure; a model is sufficient as soon as its rMSRE-V is less than 5%, so that the procedure stops increasing the model complexity as soon as the rMSRE-V reaches this threshold value. The fact that the same threshold also appears on the test proves that the training and validation distributions are indeed representative of the overall distribution.

However, the MXRE on the validation and test sets are quite different. Their means are 18.1% and 12.3%, respectively (i.e., it is smaller on the test set), and conversely, their maximum values are about 700% for the validation set and about 400% for the test set. The distribution of the relative errors of the 62,749 nontrivial models on all the 1557 test profiles has been plotted (see Fig. 7); 99.5% of the absolute values are less than 20%, 97.7% are less than 10%, and 89.2% are less than 5%.

B. Model Size and Computation Time

Even for the nontrivial relations, the models are generally small: polynomials have a mean size of 2.4 monomials and the mean number of hidden neurons equals 6 (Fig. 8).

Table 2. Model Structures and Performance

	Fulfill Accuracy Specifications	Do Not Fulfill Accuracy Specifications	Total
Polynomials	62,598	4	62,602
Neural networks	118	29	147
Total	62,716	33	62,749

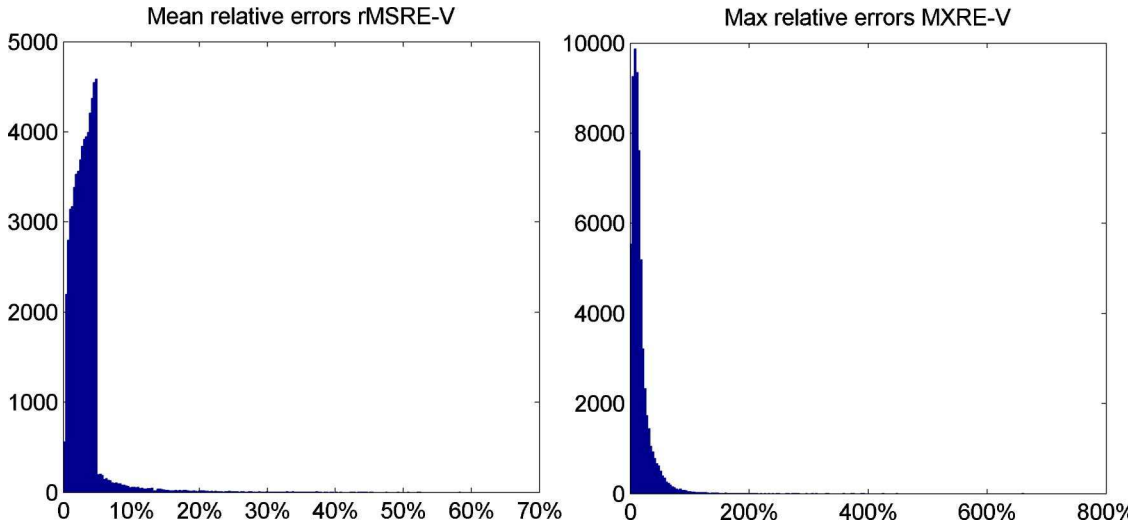

Fig. 5. (Color online) Histograms of the root mean square relative errors (left) and of the maximum relative errors (right) of the nontrivial models, on the *validation* set.

The exact computation of the CK coefficients with MATISSE for a profile, that is of the $24,300$ wavenumbers $\times$ 49 pressure values $\times$ 17 quadrature points $= 20,241,900$ CK values, requires 10 min. For a whole scene including 10,000 profiles, the computation would take about two months and is, therefore, not feasible.

How long would it take if the regression models were used instead of exact computations? Since the modeling has not yet been done for all the 24,300 spectral values, but only for 198 of them, this computation time can be estimated by computing the CK coefficients at these 198 wavenumbers $24,300/198 = 123$ times. This assumes that the model sizes are globally the same for the unseen wavenumbers. Furthermore, it necessitates that the models are appropriately loaded during the procedure.

As a matter of fact, it is not possible to consider that all the models will be loaded simultaneously in the program memory. Thus, they have to be stored in the file system, and loaded by the program when their use is requested. The file system has been organized with one file per wavenumber, each file containing 49 pressure values$\times$ 17 quadrature points $= 833$ models. The computation of a profile is done by, for each wavenumber:

1. loading the 833 models and
2. running these models.

When the models are used for a single profile, the time devoted to the second step has been measured as only 7% of the total time. This has two practical consequences for the time estimation procedure:

1. to be reliable, it must run the two steps: for each of the 198 modeled wavenumbers, the whole procedure (steps 1 and 2) has to be run $24,300/198 = 123$ times, and

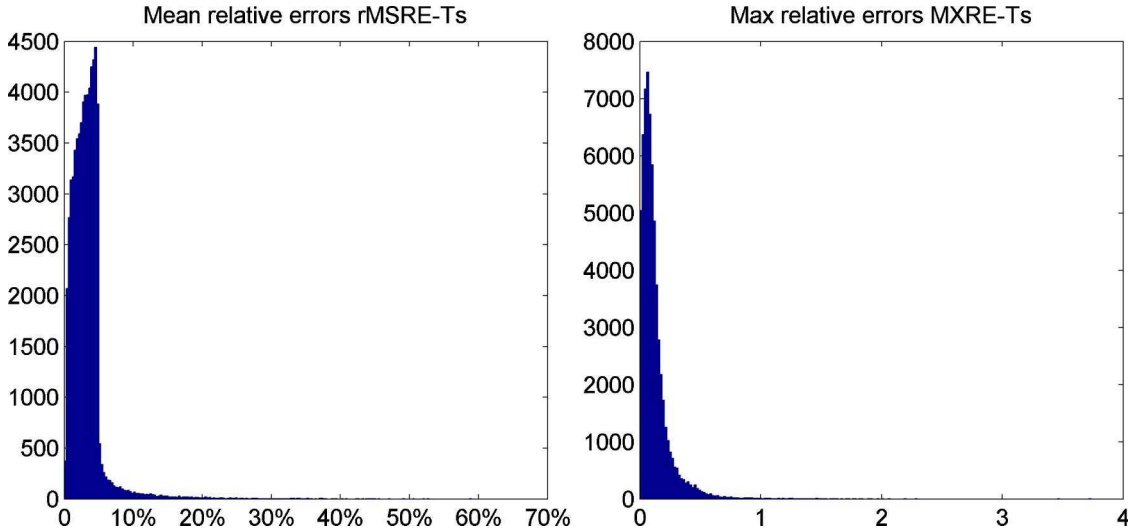

Fig. 6. (Color online) Histograms of the root mean square relative errors (left) and of the maximum relative errors (right) of the nontrivial models, on the *test* set.

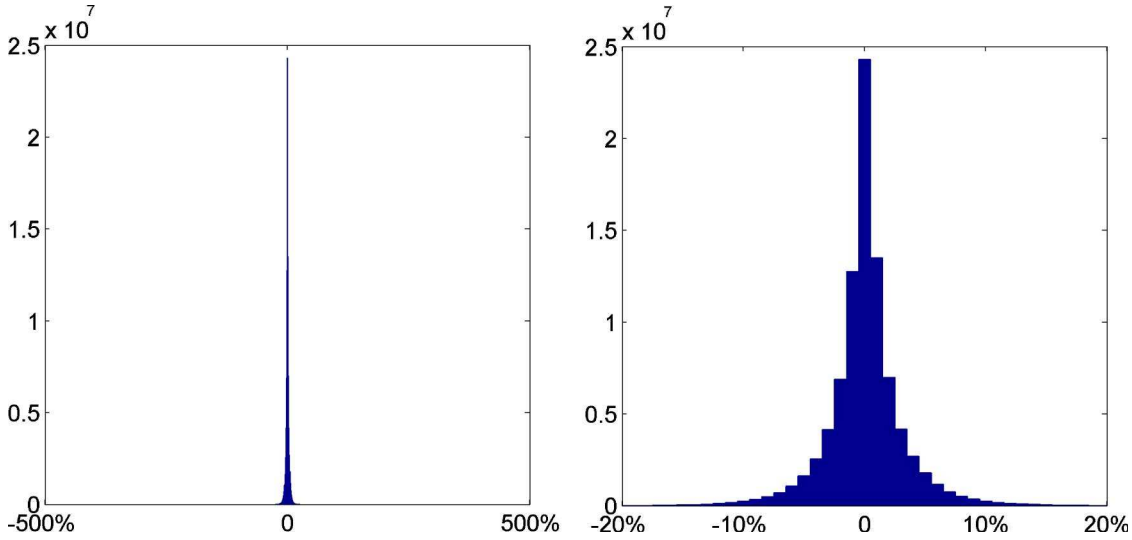

Fig. 7. (Color online) Histogram of the relative errors of the 62,749 nontrivial models on the 1557 *test* profiles. The left scale shows that the high values very seldom do, but can reach 400%, while the right scale shows that most values (99.5% of them) are smaller than 20%.

2. when the number *npro* of profiles is greater than 1, it is important, once a model has been loaded, to use it for all the profiles. Then, for each wavenumber, the first step is executed only once, and only the second step is executed *npro* times. The time needed to compute the *npro* profiles is less than the product of *npro* by the time needed to compute one profile.

This procedure has been executed on a computer with a Dual Core AMD Opteron 2212 processor (2,000 MHz). The time needed for the computation of a single profile is estimated at 1 min 15 s, which is to be compared with the 10 min needed for the exact computation. For a whole 3D scene, that is, for 10,000 profiles, the computation time is estimated at 16 h, to be compared with the two months in the case of exact CK computations; it is now conceivable to deal with such a scene.

6. Discussion

The first point we would like to discuss is the validity domain of the models, which is basically the area of the input space that is covered by the dataset. These data have been built in order to explore the whole region of interest; the geographical as well as the time localizations run through all the latitudes and periods of the year. Hence, most of the new thermophysical profiles are likely to be close to some of the learned ones. However, it is still possible, for a new image made in a particular climate condition, to have new profiles falling out of the domain. We

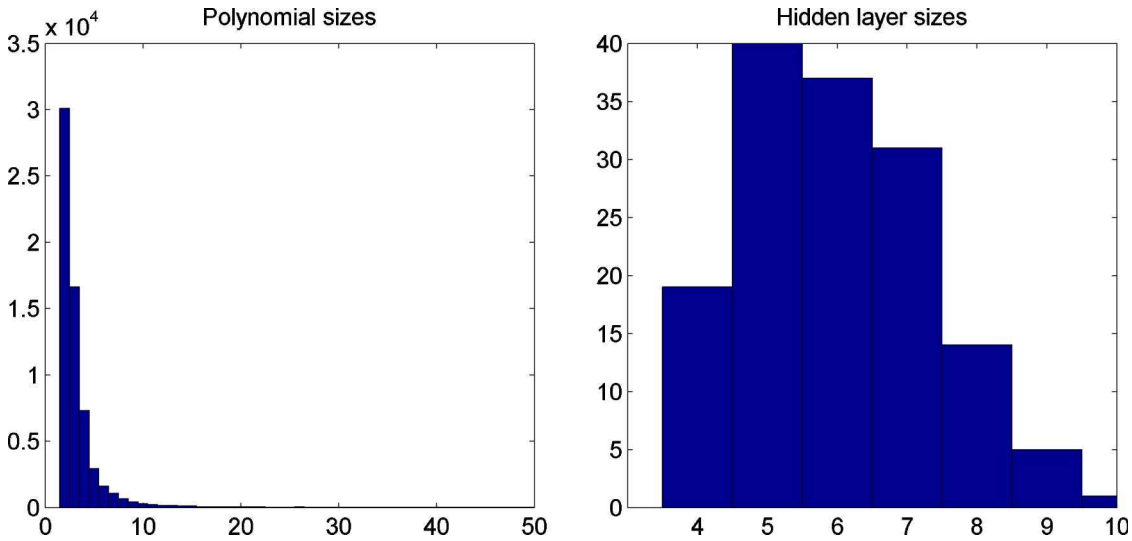

Fig. 8. (Color online) Histogram of the numbers of monomials for the 62,602 polynomial models (left), and of the hidden layer sizes for the 147 neural models (right). The number of monomials has a mean of 2.4 and runs from 1 up to 50. The hidden layer size runs from 4 up to 10 with a mean value of 6.

should, therefore, be able to detect this situation. A first safeguard consists in storing, in addition to the models, the minima and maxima of each input component in the dataset. It is very easy and fast to compare each new input vector to these values and to detect when it is out of range. But the domain delimited by these extrema is a hypercube that contains, but is not, the validity domain. The latter is very different from a hypercube; it is not even convex, so that only a part of the out-of-range profiles will be detected this way. Another possibility is to compute a confidence interval for the output. Its computation is well known in the case of a polynomial, which is linear with its parameters (see, for example, [15]), and can be obtained in the case of a neural network by using a linear Taylor expansion of the model output [16]. It should be noted that the notion of confidence interval is statistical and does not exactly apply here, since the data are nonrandom. However, it still has an intuitive signification and, also, the appealing property of getting larger and larger as the new input vector goes away from the training domain. Thus, for each new input, it gives useful information about the model validity. The computations are then more time consuming (nearly 2 min for a profile and two days for a whole 3D scene if the confidence interval is computed for each model and each profile), but are still much shorter than the exact ones. The question to decide when the confidence interval should be computed remains open.

A second point to discuss is the problem decomposition that has been chosen: for a given wavenumber and a given CK coefficient. A different local model has been built for each of the 49 discrete values of the pressure. Another option was to build a single global model for all the altitudes, taking the pressure as an additional input. In fact, this option has been implemented, too, but the relations are much more complex and the final models are, hence, very large; most of them are polynomials with more than 40 monomials. When using them in the CK computation, the time devoted to run them on a profile, relative to the time requested to load *and* to run them, enhances from 7% to 69%. The total time enhances to 2 min 15 s for a single profile, and to nine days for a whole 3D scene. This highlights the importance of finding the good decomposition of a problem before considering its modeling.

Finally, we would like to stress how useful the preliminary work has been. First, a representative dataset has been built, which covers the domain on which the models will be used. Second, thanks to a sensitivity analysis of the MATISSE computations to the CK accuracy, the required accuracy could be defined. It has been found that only the first eight nonzero CK coefficients have to be computed accurately, allowing us to focus the effort on the most crucial relations. And third, a good decomposition of the problem had to be found.

7. Conclusion

The feasibility of fast emulation by linear and nonlinear regression of the exact computation of the CK coefficients has been demonstrated. The implementation of the proposed procedure on a subset of 198 values of the wavenumber shows that a mean relative error of 3.3% and a high reduction of the computation time can be obtained: from 10 min down to 1 min 15 s, in the case of a single profile, and from two months down to 16 h in the case of 10,000 profiles. It will, therefore, be possible to deal with 3D scenes with inhomogeneous conditions, which was, up to now, inconceivable.

The proposed procedure may be applied in many other areas. As soon as a heavy numerical procedure has to be repeated many times, its emulation by fast regression models may be considered. The initial investment lies in a proper decomposition of the task to be emulated, in the specification of the requested accuracy, in the construction of a representative dataset, and, finally, in the model construction. If this preliminary work is done properly, the investment can be very rewarding, as illustrated by the performance reported here.

This work has been financed by the Direction Générale de l'Armement / Direction des Systèmes d'Armes / Unité de Management Naval.

References

1. P. Simoneau, K. Caillault, S. Fauqueux, T. Huet, L. Labarre, C. Malherbe, and B. Rosier, "MATISSE-v1.5 and MATISSE-v2.0: new developments and comparison with MIRAMER measurements," Proc. SPIE **7300**, 73000L (2009).
2. Q. Fu and K. N. Liou, "On the correlated *k*-distribution method for radiative transfer in nonhomogeneous atmospheres," J. Atmos. Sci. **49**, 2139–2156 (1992).
3. A. A. Lacis and V. Oinas, "A description of the correlated k distribution method for modeling nongray gaseous absorption, thermal emission, and multiple scattering in vertically inhomogeneous atmospheres," J. Geophys. Res. **96**, 9027–9063 (1991).
4. L. Ibgui and J. Hartmann, "An optimized line by line code for plume signature calculations—I: model and data," J. Quant. Spectrosc. Radiat. Transfer **75**, 273–295 (2002).
5. A. Berk, G. P. Anderson, P. K. Acharya, J. H. Chetwynd, L. S. Bernstein, E. P. Shettle, M. W. Matthew, and S. M. Adler-Golden, MODTRAN4 User's Manual (Air Force Research Laboratory, Space Vehicles Directorate, Air Force Materiel Command, Hanscom AFB, MA 01731-3010, 1999), p. 7.
6. G. Li, Z. Lu, G. Guo, and K. Huang, "3D atmospheric modeling based on Modtran4," WSEAS Trans. Syst. Control **3**, 483–492 (2008).
7. P. Koepke, M. Hess, I. Schult, and E. P. Shettle, "Global Aerosol Dataset GADS," Report 243 (Max-Planck-Institut für Meteorologie, Hamburg, 1997).
8. F. Karcher, "Détermination des profils atmosphériques de référence pour la mesure de constituants par spectroscopie d'absorption," Programmes PRFL et RMEL, Notes Internes CNRM (1990).
9. L. Rothman, D. Jacquemart, A. Barbe, D. Benner, M. Birk, L. Brown, M. R. Carleer, C. Chackerian Jr., K. Chance, L. H. Coudert, V. Dana, V. M. Devi, J.-M. Flaud, R. R. Gamache, A. Goldman, J.-M. Hartmann, K. W. Jucks, A. G. Maki, J.-Y. Mandin, S. T. Massie, J. Orphal, A. Perrin, C. P. Rinsland,

M. A. H. Smith, J. Tennyson, R. N. Tolchenov, R. A. Toth, J. Vander Auwera, P. Varanasi, and G. Wagner, "The HITRAN 2004 molecular spectroscopic database," J. Quant. Spectrosc. Radiat. Transfer **96**, 139–204 (2005).
10. I. Rivals and L. Personnaz, "MLPs (mono-layer polynomial and multi-layer perceptrons) for nonlinear modeling," J. Mach. Learn. Res. **3**, 1383–1398 (2003).
11. S. Chen, A. Billings, and W. Luo, "Orthogonal least-squares methods and their application to non-linear system identification," Int. J. Control **50**, 1873–1896 (1989).
12. R. Fletcher, *Practical Methods of Optimization* (Wiley, 1987).
13. A. R. Barron, "Neural net approximation," in *Proceedings of Seventh Yale Workshop on Adaptive and Learning Systems* (Yale University, 1992), pp. 69–72.
14. E. D. Sontag, "Neural networks for control," in *Essays on Control: Perspectives in the Theory and its Applications*, H. L. Trentelman and J. C. Willems, eds. (Birkhäuser, 1993), pp. 339–380.
15. G. A. F. Seber, *Linear Regression Analysis* (Wiley, 1977).
16. I. Rivals and L. Personnaz, "Construction of confidence intervals for neural networks based on least squares estimation," Neural Networks **13**, 463–484 (2000).